

ACE
ATARI
COMPUTER
ENTHUSIASTS

3662 Vine Maple Dr. Eugene OR 97405

DECEMBER 1984 - JANUARY 1985
Editors: Mike Dunn, Jim Bumpas, Larry Gold



News and Reviews

by Mike Dunn, co-Editor

The big news, as I am sure you all know, is the Atari 800XL has dropped in price to \$120 and the 1050 drive to \$180 — only \$300 for a complete disk-based computer system! Atari plans on selling 500,000 during the holiday season. This new Atari, I hear, is not exactly like our old 800XL's. They are disposable — no sockets, everything soldered in, and if anything goes wrong during the warranty period, they just give you a new one. In January, at the CES show, will be announced all kinds of new 16 and 32 bit computers, to compete with the Macintosh for under \$1000.

We have several new disks to end out the year. **Best of ACE #10-Games** is filled with most of the games since the July issue. We have not been able to make a "run-time" version of "Double-bounce" by Stan Ockers, but the rest is there, including run-time Action! games. **Best of ACE #11-Utilities** includes all the utility type programs since July, plus the ultimate version of Stan Ocker's "TinyText" with a 80 column display, done by Neville Williams from Australia and with complete documentation. From the Jacksonville ACE group, a well documented disk filled with Modem programs, including one allowing you to download programs from CompuServe with an Atari Modem — but not from Bulletin Boards. Please read Ron Ness' article about the exchange library.

Last month, I reviewed **BossXL** (Allen Macroware, POB 2205, Redondo Beach, CA 90278, \$80), and have now received their new board allowing you to switch to either the BossXL or to the original OS. Works well, now included in the package. PrintWiz by Allen macroware is what we use to dump the pictures for the newsletter, and **Typesetter** (DataArts, POB 1613, Troy, NY 12181) makes all the fancy titles you see on the listing pages. Both are excellent products.

Gary Furr, the author of the print drivers for **AtariWriter**, will sell you for \$10 the driver for your printer. He has drivers for the EPSONS, Okidatas, IDS, Gemini, Mannesman-Tally, C. Itoh and Nec., Silver Reed, etc. Write Gary at POB 1073 Mountain View, CA 94042.

This month, we received two books I can highly recommend to the intermediate to above Atari Owners. Both are similar to previous books, but cover new ground.

The **Master Memory Map** by Craig Patchett and Robin Sherer (Reston, \$16). The first memory map for the Atari was done by Robin Sherer when he owned Education Software. The first edition was just a xerox'ed sheet of locations and their uses; but was quickly expanded to a 30 page, breezy and easy to understand booklet. Last year, lad Chadwick came out with a 194 page book, **Mapping the Atari** (Compu!, \$15) covering every memory location quite well but not as entertaining as Robin's book. Now we have the latest, **The Master Memory Map**, with 361 pages, covering everything, and is a real joy to read. It has 13 appendices, 67 figures, and covers the XL computers. You'll love it.

ABC's of the Atari Computers by Dave Mentley (DataMost, \$15, available from D.E.M. POB Box 325, El Cerrito, CA 94530 with \$1.50 shipping). Written by the past president of ABACUS (Atari Users Group in San Francisco), this is an encyclopedia of useful information about the Atari. Much of it is gleaned from the best of the user group information, it includes many useful programs and hints. Do you want to know how to modify your Axlon Ramdisk so it won't crash with some software? What really is a "Framing Error"? Want a program to analyze a binary file for the RUN and INIT address? How about the printer codes for a large number of printers? It is all here and very well written. The book is similar to the also very excellent **The Atari Users Encyclopedia** by Gary Phillips and Jerry White (\$20, The Book Company), but covers different material and has different programs.

ACE BAZAAR

This year our December meeting will include a Bazaar at which members may sell or trade their hardware and software with each other. To participate as a seller, set up your wares on a table in the rear of the room, so as not to interfere with the meeting program. The meeting agenda will be short this month to allow time to browse among the tables.

Software must be original and accompanied by the original documentation to be eligible for sale at the bazaar. Members are sure to find bargains here, and sellers will have some extra money to try some new software. Before you leave the meeting, please see our VP, Larry Gold, to give him 10% of your gross proceeds. He will make a note of your name, address and the amount of your donation to WISTEC. WISTEC will send you a receipt you may use for tax deductions on your tax returns. Have fun!

CHRISTMAS PICKS

by the Editors

Hardware: The price of an Atari 800 XL is now officially at \$120. The price is lowered in anticipation of a more cheaply produced model next year. If you buy quickly, you might get one produced to sell originally at \$399. The Indus GT disk drive seems to be the most popular double density drive on the market. I've used one for a year and love it. You have your choice from among a number of good printers in the \$300 price range. For a couple hundred more you can get near letter quality. You can also get real letter quality with low cost daisy wheel printers for less than \$500. The Abati seems to be one of the better built of these low priced daisy wheel printers. Microbits now has an 1100E modem which will switch an incoming call to the handset to ring until you pick it up if it doesn't receive a carrier detect from a computer at the other end. A nice feature! The best monitor on the market now seems to be a 13" Mitsubishi. But it's very expensive, close to \$500! Maybe the technology will be used in less expensive models soon.

Software: Atari users have their choice of several excellent word processors which are as good as many used on IBM machines. The higher priced Writer's Tool by OSS, and Letter Perfect by LJK are the top of the line. But AtariWriter and Letter Wizard are also full-function processors and can be purchased for half to one-third the price. Letter Wizard is also being marketed with Spell Wizard in a single package for under \$70. Letter Wizard is the only one of the four which will not work in double density.

The Syn-Apps packages from Synapse Software are also top-featured programs. SynCalc is the best spreadsheet program I've seen for an 8-bit computer. SynFile+ is also the best filing program I've seen for 8-bit machines. These programs are compatible with each other, and they take advantage of virtual memory features permitted with the use of Axlon and Mosaic ram disks. How about a 128k spreadsheet file on an Atari? SynComm is the most full-featured communications program on the market now, although a review in the current Analog magazine describes HomeTerm with many of the same features. SynCalc and SynFile+ are less than \$70. SynComm is under \$40.

This has been a difficult year for new arcade games. The best new one I've seen is Firebug which appeared in a recent issue of Analog magazine. Family games are dominated by the dozen or so faddish trivia games. Many people are having a great deal of fun with these.

A lot of attention in software development for the Atari lately has been directed towards various "translator" systems. This will become less important the longer the XLs are on the market. But for now, a significant portion of Atari software requires a translation to run successfully on the XL. The BossXL seems to be one of the best of these for your use.

Infocom continues to come out with the best text-adventure games. And Epyx has one of the better sports games with their **Summer Games** package. We haven't seen their Baseball game, but they advertise it as the "best". Electronic Arts has a wild and popular game in their **One on One** basketball game.

In strategy games, Electronic Arts has a winner with **Archon II — Adept**. Strategic Simulations has a real challenge for board wargamers in their **War in Russia**.

U.S.Adventure

Although billed as an adventure type game, we found Jerry White's U.S.Adventure by First Star Software the perfect addition to the family collection of quiz games. The adventure part of this game is the unique approach to the question format accomplished by the use the joystick to pick the course of the game.

The educational objective of this game is to teach the history and geography of the United States. We feel it does this in an interesting way. The object of the game is to acquire as many points as possible. The user travels from state to state in the order the states were admitted to the union by correctly selecting the year in which events in U.S. history occurred. The children especially liked the "time tunnel" used to display the dates for selection. When a choice has been made the user is told either that the event occurred in the year selected, or that the event took place before or after the year specified.

We find this game challenging and fun to play as a group. We especially liked the "trivia break" which appears once every ten correct events. If your family is into history or the current trivia and quiz game fad, this is a game not to be missed.

— Ruth Ellsworth

BUMPAS REVIEWS

GULF STRIKE (Avalon Hill, \$30) is a one or two-player strategy game simulating a hypothetical operational scenario pitting the USSR and Iraq against Iran, supported by the US, Britain, France, Kuwait, Saudi Arabia and the Gulf States. The game lasts 25 turns, and will end as soon as the USSR player captures 9 of the 21 victory point spaces. The turns are divided into 3 Phases: Surface movement; Air movement and combat; and Surface combat. The first turn or so might take players an hour or more to complete. But after a few turns, the time required drops dramatically to as little as 7 or 8 minutes for the last turns.

The map is very large, smoothly scrolling across more than 24 screens, containing 17 different types of terrain features. One type of terrain seems to have no purpose in the game — seaports. The playing area covers geography containing more than 100,000 km sq., divided into a 55x65 grid of spaces. Iran, Kuwait, Bahrain, Qatar, and parts of the USSR, Iraq, Saudi Arabia, and the United Arab Emirates are portrayed on the screen. The players control units differentiated among over 30 types, all with varying characteristics.

The screen is divided into 3 windows. The top contains one screen of the map — an area of 8x18 spaces. Below that is a graphics window in which up to 19 of the units in a stack are shown, along with the terrain under the stack. A player may only perform operations on these 19 units in a stack. All other units in the stack are off-screen. You might find it advisable to keep stacks below this 19 unit limit for maximum flexibility.

At the bottom is a 5-line text window displaying unit characteristics and status of the unit selected. A unit is selected by placing the cursor over a stack on the map; then over the unit symbol in the graphics window below the map. Then a combination of joystick directions and button pushes permits the player to select from among 6 attack or defense formations, and to move the unit. An entire stack may also be moved, unless there is a naval unit in the space. A naval unit must be moved individually (why?), and no stack may move until all naval units are first moved out.

Air combat occurs when you send a strike of one or more aircraft to a space containing an enemy stack. The strike might be intercepted if the enemy has aircraft assigned to intercept, and the range and radar permit. The success of radar seems to be somewhat random.

Surface combat occurs between adjacent stacks of enemy units. If a unit is adjacent to more than one enemy stacks, it attacks the unit with the lowest space number. During combat the map is scanned across the screen from the upper left corner. If you can position your units above and to the left of enemy stacks, you will have an advantage by attacking first. Combat results include only loss of hit points and elimination. Air strikes may only reduce a target to 1 hit point, but naval bombardment may eliminate units.

All player input is by joystick, except for pushing the Start key at the end of each phase. The large, multicolor map is a wonder, coming close to the standard set by Chris Crawford in his games, although the colors are a bit drab. The programming seems very sophisticated, permitting wide player flexibility without getting in the way of play. The only indication of a failure by the programmer is the need to hold down the Option key for 40 seconds or so while the program loads on an XL machine. A poke could have taken care of this without making the player guess how long the key must be held down. I look forward to seeing this game system developed in future scenarios. Meanwhile, Gulf Strike will give you several hard-fought play-throughs solitaire. You might never run out of combinations of game strategies in the two player game.

IMPERIUM GALACTUM (Strategic Simulations, Inc. \$40) is a game of space exploration and conquest for 1 to 4 players. Each player begins the game at the dawn of space exploration on their home world. Players control population, the number of mines and industry and farms on controlled worlds, and after you've raised your technology level you may also improve the environments on controlled planets, permitting greater population growth. The winner is the player with the largest population when you decide to cease playing.

Exploration is conducted with transports and warships. Ships are built by players from a menu of 8 warships, divided among small, medium and large classes. A warship class may also be redesigned and renamed. I was unable to successfully redesign any but a small class of ship. I couldn't get medium and large classes to take a speed characteristic and thereby exit the design mode of the program. Players may also build trading ships to transport goods between planets. Transport ships carry only armies and colonists.

One very unique feature of this program is the ability to conduct diplomacy with the computer players. The negotiation menu shows your attitude (friendly, neutral or hostile) towards the other players, and theirs towards you. You may freely change your own attitude, but the computer players will mostly refuse your efforts to change theirs towards you. They will also mostly refuse your efforts to get one or more of them to help you by attacking another player (computer or human). But when they accept your offer, it's extremely gratifying. Another menu permits you to negotiate with independent worlds, which can also be valuable allies.

Warships must be assigned to a task force and given one of 6 missions, including Garrison, Repair, Patrol, Commerce Raid, Embargo and Move. Combat might occur at the end of one of the two move phases each turn. Combat may occur between warships, or between ships and planets. Enemy planets may be captured, or just purged of all civilization.

Most of the game is played on the series of text screens. There is a strategic map, a 20x40 dot grid showing all the star systems. A system may have from 0 to 2 habitable planets. There is a tactical map showing approximately 1/4 the strategic map, with star systems and task forces displayed in larger size. Friendly task forces are shown, along with the presence of task forces of other players in systems containing friendly task forces.

The game moves very quickly and is fun to play. No one has to look away from the screen for very long while another player does a portion of a turn. The phases are broken up so long periods of inactivity are avoided. The multi-player feature and diplomacy with the computer players make this a fun game of many possibilities in play and outcome.

Stealth (Broderbund) is a 5-level 3D arcade game. You are the pilot of a Stealth bomber trying to fight through to the enemy tower and destroy it. The tower shows on the screen so you can see it as you approach. Enemy ground batteries and aircraft shoot you on the way in. You may also increase energy by flying through energy fields. This will extend your range and help you reach your goal.

The game is played with the joystick and you control speed and direction. Altitude is constant. This is an improvement on the Zaxxon-type game. Get your joystick muscles limbered up!

WHISTLER'S BROTHER (Broderbund) is an action adventure game reminiscent of Hard Hat Mack. Your quest is to guide your brother over territory he covered the year before, to pick up all the chapters to his book written for credit at the university. You can find tools to help you in your quest and to fight off various beasts who will try to stop you.

An interesting feature of this game is you not only have to look out for yourself, but you must also protect your brother. If he runs into any hazard, it will end your turn as quickly as if you ran into it yourself. The only control you have over him is your "whistle" (done with the joystick). You must whistle to keep him nearby. If he wanders off, it's danger for you!

The graphics and sound are excellent in this program. The scenes of each chapter are very well and interestingly drawn. This game will give you hours of fun.

— Jim Bumpas, co-editor

VP's RAMBLINGS

All the indications point to an upsurge in Atari use by the public. The next few months should see people buying more and more Ataris and the other things going with it; i.e., disk drives, printers, modems, etc. Since Atari seems to be dropping the price on everything and making owning a computer affordable to just about everyone, the user groups should be prepared to help the new users. We have started a new group of people called "Bugbusters". These are people all over the country who have special skills and are willing to help those with problems. You will find the first names and phone numbers and addresses in this issue of the newsletter, also these names and numbers will be listed on the BBS in a special section called "Bugbusters" of course.

The BBS is going along just fine and we are sorry we had to erase all the old passwords and make you call in again so we may bring all members up to level 7 to take advantage of the new programs for members only. Remember: if you are a member please call and sign in again. Thanks.

Since this is the holiday season I want to wish all of you the best and may all your wishes come to pass and may you have nothing but happy computing with no errors.

— LARRY GOLD

XYGRAF

PART 1:

XYENTRY. I've occasionally had need of a program to produce a simple X-Y graph. I thought certainly there must be a program in all my back issues of magazines. Well, I couldn't find anything I liked, so I decided to write my own. 'XYENTRY' is the first part of this process. It allows you to enter data pairs, modify them, save them to disk or recall them as necessary.

Run the program and choose option 1 to enter or modify data. An orange cursor shows where the next X or Y value will be entered. A summary of the commands is listed at the bottom of the screen. All alphabetic one letter commands require the CTRL key be pressed along with the letter. Here is a further description of each of the commands:

CTRL (arrows). (move the cursor). The cursor keys move the cursor to valid positions on the page. Enter a number to have it occupy the position. Numbers can have up to 15 characters, (only 9 or 10 digits are valid). The cursor will automatically advance to the next position, the next page if necessary. Once entry is started, a blinking cursor reminds you entry has to be completed. Not every position need be occupied, a DELETE will clean up unoccupied lines.

CTRL M (Mark a pair). When operating with a set of data you often wonder what will it look like to remove this pair or that one. You could delete them from the set but you would have to type them in again to replace them. Instead, provision has been made to mark the pair at the current cursor position with an asterisk. Pairs so marked will not be used in calculations or appear on graphs. To make the pair valid again, use the Undo command. Marked data does appear when listed and is marked with an asterisk.

CTRL D (mark a pair for Deletion). You can replace a number by entering another in it's place but to get rid of a data pair you must first mark it for deletion and then do a 'DELETE'. CTRL D will put a 'D' in the mark column. The pair will be removed at the next operation cleaning up the data.

CTRL U (Undo). To take the asterisk off a data pair at the current cursor location, use the Undo command. If the pair is marked for deletion, the Undo will undo the deletion mark.

CTRL N (Next page). If there is a next page available for data, it will be displayed. The number of pages is equal to the total available data pairs divided by 20. The number of pairs is set at 100 in line 20 (MPTS = 100). Fewer or more pairs can be allowed for, (multiples of 20 are recommended). Larger numbers tend to slow down the cleaning up process considerably.

CTRL B (Back up a page). To review data on previous pages (or change it) you can back up.

CTRL X (eXchange data). Use this command to switch data in the X and Y columns. Only the X column can be ordered.

CTRL R (Read a file from disk). Data can be entered from the keyboard, read off the disk, or a combination of both. Data from the disk will be added to data already in the set. If you want only data from the disk, you must do a CLEAR before asking for the data.

CTRL S (Save file on disk). Files are saved with the extension '.DAT'. If necessary, a cleanup and deletion of data will be done before the file is saved.

CTRL O (Order the data). For some operations like graphing the data must be ordered. The ordering is done on the X column of values — a cleanup of data precedes the sorting.

CTRL E (write Equation to change values). Suppose you want to graph $\log(Y)$'s rather than Y's. Using this command you can write a Basic line (line #980) which is inside a loop indexed by J running over the available data. The line $Y(J) = \text{LOG}(Y(J))$ changes all Y's to $\log Y$'s. When prompted with the 980, you can write any valid Basic line using the variables X(J), Y(J) and J. It is possible to write incorrect lines or lines which will not work with some values (taking the square root of a negative number for example). If you get an error message, you should check your data again as it may have been partially transformed. If you ask for this option when there are no entries, it will be assumed you wish to create X values using an equation. You will be asked for the maximum entry you want to create. For example, to set up 40 values of X in increments of 5 you could set the maximum to 40 and enter $X(J) = J*5$.

ESC (ESCAPE to menu). Use this command to leave the data entry-modification program and automatically load other programs.

DELETE (Delete lines marked for deletion). A clean up of data will take place removing all unused lines and eliminating those marked 'D'.

CLEAR (Clear out all data).

CTRL F (list Files from disk). Only .DAT files are listed.

CTRL T (sTatistics). Simple statistics are given for column Y.

CTRL L (List data on printer). Values are listed in E format. The variable NDP in line 30 controls the number of decimal places. Numbers are truncated after the last decimal point, not rounded.

This is only the first part of 'XYGRAF'. The part that does the actual graphing should be finished by next issue. I welcome comments and suggestions, I know there could be some improvements. I'm aware of the flickering when numbers are entered. It has to do with the display list interrupt routine using WSYNC which is affected by the keyclick routine. De Re Atari recommends using VCOUNT or rewriting the keyboard handler. I haven't had time to try either of these solutions yet.

— Stan Ockers
R.R. #4, Box 209
Lockport, IL 60441

FROM THE LIBRARY

First of all, I want to apologize to all of you who have sent for disks and library lists for the slow service you have received during the past 3 months. I assure you we will do our best to improve our service to you as things smooth out over the next couple of months.

I also feel an explanation is in order. The main reason for this is my employer for the past 24 years decided to discontinue serving Eugene. This left me and my family in a difficult position. We had only a very short time to decide whether to stay in Eugene where we have lived all of our lives or to accept a company transfer to a location we do not feel is as desirable a place to live.

I have decided to give up a career in the airline industry and take up a new career. Just what this new career will be, I don't know. The main thing is we have now made the commitment to stay in the Pacific Northwest. If any of you know of any openings in the airline industry in station operations, dispatch, cargo service, etc., let me know.

Anyway, back to the library . . . by the time this newsletter reaches you, a new library list should be ready. New material has been coming in and by the time we have it sorted and checked, the library list is out of date. The target date to have the list off the press is December 1. This date may slide a bit.

Several people have asked what we do with a program when someone sends one in for the newsletter or library exchange. I thought I would pass these tips along to you. Perhaps they will save us all some time. The first step when I receive a program is to make a back up copy. I learned this the hard way a long time ago.

The next step is to try and check the various directories to see if this program duplicates a program in the library or is something which could be covered by copyright laws or perhaps is an improved version of something in the library.

The third step is to run the program and see exactly what the program does, and then I deliberately try to make it crash. Is the program properly TRAPped? What happens if false information is entered? If the program is set up for use with a printer, what happens if the printer is not turned on? Can I get back into the program or will it crash or lock up?

Assuming the program passes these tests, I then pass it along to the editors of the newsletter or the educational groups for them to try. If the program does not pass the tests, I usually will try and fix it or ask someone in the club to take a look at it or even send it back to the author. The better your program is, the better we like it. Documentation on the disk is greatly appreciated too.

PRICE ADJUSTMENTS — Effective with the new library list, we will be adjusting the price of all disks to a flat charge of \$10 single sided and \$15 double sided. This is a slight increase for some disks and a reduction for others. This price includes postage and handling to all addresses in the USA, Canada, and all APO and FPO military addresses. Outside the USA and CANADA please add a flat charge of US\$2.00 for airmail service. This is a flat charge regardless of how many disks you order whether it be 1 disk or 50 disks being sent to addresses outside the USA and Canada.

Until next year, HAPPY HOLIDAYS from the librarians and the rest of the Ness family.

— Ron Ness

DIFCONV by David Fuller

check Oct. issue for text

```

10 TRAP 888:GOSUB 900:PRINT CHR$(125);
TTL$
20 REM ** GET FILE NAME
30 PRINT "          by David Fuller"
40 PRINT :PRINT " How may drives (1 or
2) ";:INPUT DRIVES
50 IF DRIVES<1 OR DRIVES>2 THEN 40
60 POSITION 2,6:PRINT "1. Filemanager"
70 POSITION 2,7:PRINT "2. Other"
80 PRINT :PRINT "Enter Type of File ";
:INPUT ANSR$
90 IF ANSR$<"1" AND ANSR$<"2" THEN 6
0
100 IF ANSR$="2" THEN TYP=2:GOTO 710
105 TYP=1
110 PRINT :PRINT " Insert Filemanager
Data Disk"
120 PRINT " in Drive 1 and press RETU
RN";:INPUT ANSR$
130 PRINT CHR$(125);TTL$:PRINT :PRINT
" Files on the Disk":PRINT LIN$
140 INFIL$="D1:*. *":OPEN #1,6,0,INFIL$
150 INPUT #1;FLNAM$:IF FLNAM$(5,8)="FR
EE" THEN 180
160 IF FLNAM$(11,13)="DAT" THEN PRINT
FLNAM$(1,8);" ";FLNAM$(14,17)
170 GOTO 150
180 CLOSE #1:PRINT LIN$:PRINT "Which f
ile to convert ":PRINT " ";:INPUT
ANSR$:INFIL$="D:":INFIL$(3)=ANSR$
190 INFIL$(LEN(INFIL$)+1)="*.FMT"
200 REM ** GET AND DISPLAY FIELDS
210 PRINT CHR$(125);TTL$:POKE 752,0
220 OPEN #1,4,0,INFIL$
230 INPUT #1;NUMFLDS
240 INPUT #1;FLDN$
250 MAXLEN=0
260 FOR N=1 TO NUMFLDS:INPUT #1,A:FN(N
)=A:IF A>MAXLEN THEN MAXLEN=A
270 NEXT N:A=0
280 CLOSE #1:LE=LEN(INFIL$):INFIL$(LE-
3,LE)="IDK":OPEN #1,4,0,INFIL$
290 INPUT #1,A:NUMR=A
300 FOR N=1 TO 4:INPUT #1;IDK1:NEXT N:
INPUT #1;IDK2:INPUT #1;IDK3
310 PRINT :PRINT "File ";CHR$(34);INFI
L$(3,LEN(INFIL$)-4);CHR$(34);" has ";N
UMR;" records"
320 PRINT :PRINT "Field names"
330 PRINT LIN$:A=0:B=5:C=3
340 FOR N=1 TO NUMFLDS*12 STEP 12:A=A+
1
350 B=B+1:IF A=11 THEN C=20:B=6
360 IF A=10 THEN C=2
370 POSITION C,B:PRINT A;". ";FLDN$(N,
N)
380 NEXT N:POSITION 2,16:PRINT LIN$
390 PRINT " is this the right File (Y
/N) ";:INPUT ANSR$
400 IF ANSR$<"Y" THEN RUN
401 PRINT :PRINT "How many records to
copy ";:INPUT CPY
405 PASS=1:IF CPY<NUMR THEN PASS=2
406 START=1
410 INFIL$(LEN(INFIL$)-2)="DAT"
420 REM ** WRITE DIF FILE
430 PRINT :PRINT " Enter name of outp
ut file":PRINT " up to 8 characters
";:INPUT ANSR$
440 OUTFIL$="D:":OUTFIL$(2)=STR$(DRIVES
):OUTFIL$(3)="":OUTFIL$(4)=ANSR$:OUTF
IL$(LEN(OUTFIL$)+1)="DIF"
442 IF DRIVES=1 THEN GOTO 450
443 PRINT :PRINT "Insert destination d
isk"
445 PRINT " in Drive 2 and press RET
URN";:INPUT ANSR$
450 PRINT CHR$(125);TTL$:PRINT :PRINT
" Processing..."
455 POSITION 5,8:PRINT "Total Records:
";NUMR
460 POSITION 5,10:PRINT "Field: ";:POS
ITION 5,12:PRINT "Record #:";
470 OUT$=CHR$(34)
480 OPEN #2,8,0,OUTFIL$
481 IF PASS=2 AND START>1 THEN GOTO 48
6
485 CLOSE #1:OPEN #1,4,0,INFIL$:NOTE #
1,SEC,BYTE
486 IF START<1 THEN CPY=NUMR
490 PRINT #2;"TABLE":PRINT #2;"0,1":PR
INT #2;OUT$;OUT$:PRINT #2;"VECTORS":PR
INT #2;"0,1":STR$(CPY-START+1):PRINT #2
;OUT$;OUT$:PRI
500 PRINT #2;"0,1":STR$(NUMFLDS):PRINT
#2;OUT$;OUT$
510 PRINT #2;"DATA":PRINT #2;"0,0":PRI
NT #2;OUT$;OUT$
511 A=0
515 IF TYP=2 THEN ST=0:FOR N=1 TO NUMF
LDS:A=A+FN(N):NEXT N:GOTO 550
520 FOR N=1 TO NUMFLDS:FT(N)=1:FN(N)=1
4
530 IF FN(N)>100 THEN FT(N)=1:FN(N)=14
532 A=A+FN(N)
540 NEXT N:ST=0
550 IF A>255 THEN PRINT "Total record
length over 255 characters, can not co
nvert ":END
555 FOR Z=1 TO NUMFLDS
560 IF TYP=2 THEN POSITION 13,10:PRINT
Z:GOTO 580
570 POSITION 13,10:PRINT FLDN$(Z*12-11
,Z*12);
580 PRINT #2;"-1,0":PRINT #2;"BOT"
590 ST=ST+FN(Z)
600 FOR X=START TO CPY
610 POSITION 15,12:PRINT X;" ";
620 INPUT #1;INPT$
630 IF FT(Z)=0 THEN PRINT #2;"-1,0":PR
INT #2;OUT$;INPT$(ST-FN(Z)+1,ST);OUT$
640 IF FT(Z)=1 THEN PRINT #2;"0,1":VAL(
INPT$(ST-FN(Z)+1,ST)):PRINT #2;"0"
650 NEXT X
655 IF Z=NUMFLDS THEN NOTE #1,SEC,BYTE
660 POINT #1,SEC,BYTE
670 NEXT Z
680 PRINT #2;"-1,0":PRINT #2;"EOD":CLO
SE #2
681 IF CPY=NUMR THEN GOTO 690
682 PRINT :PRINT :PRINT "Do you want t
o copy the rest of the":PRINT " recor
ds ";:INPUT ANSR$:IF ANSR$(1,1)<"Y" T
HEN 690
683 POINT #1,SEC,BYTE
684 START=CPY+1
685 IF DRIVES=2 THEN PRINT :PRINT "Ins
ert new destination disk ":PRINT " and
press RETURN ";:INPUT ANSR$
686 PRINT CHR$(125);TTL$:GOTO 430
690 POSITION 5,19:PRINT " DONE ":END
700 REM ** GET OTHER FILE INFO
710 PRINT CHR$(125);TTL$:PRINT " Con
vert other file"
720 PRINT :PRINT "How many fields in f
ile ";:INPUT NUMFLDS
725 PRINT :PRINT "How many records in
file ";:INPUT NUMR:PRINT
726 PRINT :PRINT "How many records to
copy ";:INPUT CPY:PRINT
728 IF CPY<NUMR THEN PASS=2
730 FOR Z=1 TO NUMFLDS
740 PRINT "Length field #";Z;:INPUT X:
FN(Z)=X
750 PRINT "Type of field (5)tring/(N)u
mber ";:INPUT ANSR$:IF ANSR$(1,1)<"S"
AND ANSR$(1,1)<"N" THEN PRINT "S or
N":GOTO 750
760 IF ANSR$(1,1)<"N" THEN FT(Z)=1:GO
TO 780
770 FT(Z)=0
780 NEXT Z
790 PRINT CHR$(125);TTL$:PRINT :PRINT
" # Length Type":PRINT LIN$
800 FOR Z=1 TO NUMFLDS:PRINT " ";Z;"
";FN(Z);" ";
810 IF FT(Z)=0 THEN PRINT "String"
820 IF FT(Z)=1 THEN PRINT "Number"

```


difconv cont

CAI

by Jeff McHie

```

830 NEXT Z:PRINT LINS
840 PRINT :PRINT "Are all entries correct (Y/N) ";:INPUT ANSR$:IF ANSR$(1,1)
<"Y" THEN GOTO 710
850 PRINT CHR$(125);TTL$:PRINT :PRINT
"Enter name of source file":PRINT " example: TEST.DAT ";:INPUT ANSR$
860 INFIL$="D":INFIL$(3)=ANSR$
865 START=1
870 PRINT CHR$(125);TTL$:GOTO 430
880 ERR=PEEK(195):ERLIN=PEEK(186)+256*
PEEK(187):PRINT :PRINT "          ERROR:
";ERR;" AT LINE ";ERLIN
890 PRINT :PRINT " Press RETURN to
continue ";:INPUT ANSR$
895 RUN
900 REM ** SET UP VARIABLES
910 PRINT CHR$(125);:DIM TTL$(34):TTL$
=" CONVERT TO DIF FILE UTILITY"
920 DIM ANSR$(255),INFIL$(17),FLDN$(24
0),LINS(40),CHGFLD$(12),FNCT$(40),FLNA
M$(17),OUTFIL$(17),OUT$(1)
930 DIM CHG$(255),IMPT$(255),X$(1),SPY
PE$(2),FLD1$(255),FLD2$(255),FN(20),FT
(20)
940 LINS$="-----
-----":PRINT LINS:A=0:B=5:C=2
950 RETURN

```

```

0 REM FIGURE 4
10 REM C.A.I. DRIVER PROGRAM
20 REM BY JEFF MCHIE
30 PORTA=54016:REM DATA OR DIRECTION R
EGISTERS
40 PACTL=54018:REM PORTA CONTROL REGIST
ER, SELECTS WHETHER PORTA WILL BE DATA
OR DIRECTION (INPUT/OUTPUT) REG.
45 POKE PACTL,56:REM SET PORTA AS A DI
RECTION REGISTER
50 POKE PORTA,8:REM SELECT BIT 3 AS AN
OUTPUT BIT. OTHER BITS WILL NOT BE US
ED & WILL REMAIN AS INPUT BIT
60 POKE PACTL,60:REM RESET PORTA AS A
DATA REGISTER
70 REM NOW IF WE SET BIT 3 (PIKE PORTA
,8) BIT LINE 3 WILL GO HIGH, IF WE POKE
PORTA,0 BIT LINE 3 WILL GO LOW
80 REM LINES 10-70 SELECT PORTA,BIT3 A
S AN OUTPUT LINE
90 DIM C$(3)
100 INPUT C$
110 IF C$="ON" THEN POKE PORTA,0
120 IF C$="OFF" THEN POKE PORTA,8
130 GOTO 100
140 REM LINES 100-130 LET YOU TURN ON
OR OFF APPLIANCES BY EITHER TURNING ON
OR OFF BIT 3

```

DECEMBER

MEETING

WEDNESDAY

12th. 7:30pm

SOUTH EUGENE

HIGH

TEE OFF

MAGIC 1 & 2

```

92:POKE 707,222:POKE 623,1:60SUB 10:RE
TURN
32700 DATA 1,7,31,63,63,127,127,255,12
8,224,240,252,252,254,254,255
32701 DATA 255,127,127,63,63,31,7,1,25
5,254,254,252,252,240,224,128
32702 DATA 1,1,3,3,7,15,63,255,128,128
,192,192,224,240,252,255
32703 DATA 255,63,15,7,3,3,1,1,255,252
,240,224,192,192,128,128
32704 DATA 24,60,60,126,126,126,255,25
5,255,128,183,165,165,183,128,255
32705 DATA 255,255,126,126,126,60,60,2
4,192,240,254,255,255,254,240,192
32706 DATA 28,31,28,16,16,16,16,255
,255,255,255,255,255,255,255
32707 DATA 16,2,64,0,8,128,2,16,128,25
4,183,251,127,222,124,56
32766 DATA 59,62,64,67
32767 DATA 224,224,128,128,128,128

```

```

381 IF PEEK(764)<255 THEN POKE 764,25
5:RUN "D:MENU"
390 IF P<510 THEN 360
400 GOTO 370
410 FOR M=1 TO 33:NEXT M:RETURN
420 REM STRING MAGIC ROUTINE
430 A=(ADDR-PEEK(140)+PEEK(141)*256):A
H=INT(A/256):AL=A-(AH*256):VUT=PEEK(13
4)+PEEK(135)*256:Q=VUT+VNUM*8+2
440 IF A<0 THEN A=PEEK(140)+PEEK(141)*
256:GRAPHICS 0:? "CAN'T GO BELOW ADDR
55":A:STOP
450 IF PEEK(Q-2)<128 THEN GRAPHICS 0:?
"VARIABLE #";VNUM;" NOT A STRING":STO
P
460 POKE Q,AL:POKE Q+1,AH:AH=INT(SIZE/
256):AL=SIZE-(AH*256):POKE Q+2,AL:POKE
Q+3,AH:POKE Q+4,AL:POKE Q+5,AH:RETURN

```

JANUARY

MEETING

WEDNESDAY

9th. 7:30

SOUTH EUGENE

HIGH


```

1002 ? #6;" Jst JJJJJJJJJ
  yi JJJJJJKKKKK 514wi JJJKKKKKKJ
  H12 uv JJJKKKKKJ514 5t JJJJJ"
;
1004 ? #6;"KJ 1111W11 JJKJKJ 112
514v JKKJ 112";BH=48:BV=PB
+35:HH=105:HV=PB+210:FV=PB+340:RETURN
1100 POSITION 0,0:? #6;"K":POSITION 1,
2:? #6;"JJJKKKKKKKKKKKKKKKKKJJJJ
512 514KKJJ z 1111Jst "
1107 ? #6;" [ 1111W11
UK JJKJJJJ 114JJKJJ JKKKKKK
KKKKKKKK";POSITION 10,0:? #6;"N";
1104 BH=49:BV=PB+63:HH=108:HV=PB+177:F
V=PB+299:RETURN
1200 POSITION 0,0:? #6;"K":POSITION 0,
0:? #6;" KJJKKKKKKK N KJJKKJ
KsitJ KJJJ y11KK KJJ";
1202 ? #6;" uvKJJ JK J1
Jst KJ J11111 JuiKJ JK
111111 vWKJ J511112 JJS1 "
;
1204 ? #6;"KK111111 514-114 JKKKKKKK
UK 1111KKKKKKKKKKKKKKKKKK";BH=57:BV=
PB+87:HH=190:HV=PB+222:FV=PB+344
1205 RETURN
1300 POSITION 0,0:? #6;"K":POSITION 1,
2:? #6;"JJKKKKKKKKKKKKKKKKKJJJJJJ
1 514KKK 1111111 "
1302 ? #6;" 1111112
1 KKKKKKKKKJJJ UK KKKKKKKKKKKJ
KJJKKKKKJJ";POSITION 10,0:? #6;"N";
1304 BH=56:BV=PB+56:HH=190:HV=PB+180:F
V=PB+310:RETURN
1400 POSITION 0,0:? #6;"K":POSITION 0,
0:? #6;"JJKKKKKKKJJJJJJJJ K511
514KKJJJJJJ112 1 KJKwitKVV "
1402 ? #6;"JJJJ siivK JJJJJJ
K1J uiZJKJ KJKJJ1111 KJJJJ JJ
J5111111KK KKKJ 1 K11112 111 JJKJ";
1404 ? #6;"1 1111 JJ11JK
st UK JKKKKKKJJwi1KKJJ";BH=57:BV
=PB+33:HH=188:HV=PB+208:FV=PB+330
1405 RETURN
32000 DIM PR$ (50):CB=PEEK (106)-4:POKE
106,CB:PR$ (1,41)="hh,Oh,hh,TP,LD,C,H
H+LL+HH2+FH0HdTP0+""
32010 GRAPHICS 18:Z=CB*256:A=USR (ADR (P
R$),Z,4):FOR M=Z+408 TO Z+511:READ D:P
OKE M,D:NEXT M:POKE 756,CB
32020 FOR M=Z+328 TO Z+351:READ D:POKE
M,D:NEXT M:RETURN
32100 POKE 54279,CB:PB=CB*256+512:POKE
53277,3:POKE 559,46:POKE 53248,0:POKE
53249,0:POKE 53250,0:POKE 53251,0
32110 POKE 704,14:POKE 705,0:POKE 706,

```


XY ENTRY BY STAN OCHERS

```

10 REM *****
11 REM **      XYENTRY      **
12 REM **      PART OF XYGRAF  **
13 REM **      S. OCKERS 11/84  **
14 REM **      ACE Newsletter Dec 84  **
15 REM **      3662 Vine Maple Dr.  **
16 REM **      Eugene, OR 97405  **
17 REM **      $12 year  **
18 REM *****
19 REM
20 MPTS=100:DIM N$(20),D$(20),R$(20),N
BLK$(15),FNM$(14),DAT$(4),X(MPTS),Y(MP
TS),MARK(MPTS):NBLK$=""
30 DAT$=" ".DAT":MPGS=MPTS/20:GOSUB 3000
:OPEN #1,4,0,"K":GOSUB 1000:NDP=5:POK
E 16,112:POKE 53774,112
99 REM * MENU *
100 ? CHR$(125);"? " MENU"? :? "
(1) ENTER AND MODIFY DATA"? :? "(2) D
RAW AN X,Y GRAPH"
150 GET #1,K:IF K<49 OR K>50 THEN 150
160 IF K=49 THEN ? CHR$(125):GOSUB 105
0:GOTO 310
170 IF K=50 THEN ? :? "Function not i
mplemented yet":GOSUB 1370:GOTO 100
309 REM * MAIN KEYBOARD INPUT *
310 POKE 694,0:GET #1,K:N=(PAGE-C1)*20
+ROW
320 IF K=28 AND ROW>C1 THEN GOSUB 910:
ROW=ROW-C1:GOSUB 900
330 IF K=29 AND ROW<20 THEN GOSUB 910:
ROW=ROW+C1:GOSUB 900:UPDAT=C1
340 IF K=30 AND COL=VLT THEN COL=XLT:G
OSUB 920
350 IF K=31 AND COL=XLT THEN COL=VLT:G
OSUB 920
352 IF K=27 THEN GOSUB 910:GOTO 100
354 IF K=20 THEN 1700
360 IF (K)47 AND K<50 OR K=43 OR K=45
THEN 400
362 IF K=13 AND (X(N)<>N9 OR Y(N)<>N9)
THEN MARK(N)=M1:POSITION MLT,ROW:? "X
";
363 IF K=4 AND (X(N)<>N9 OR Y(N)<>N9)
THEN MARK(N)=M2:POSITION MLT,ROW:? "D"
;:UPDAT=C1
364 IF K=21 THEN MARK(N)=C0:POSITION M
LT,ROW:? " ";
366 IF K=125 THEN GOSUB 910:? CHR$(125
);" clearing data":GOSUB 1000:GOSUB 10
50
368 IF K=156 THEN GOSUB 1100:GOSUB 105
0
370 IF K=14 AND PAGE<MPGS THEN PAGE=PA
GE+C1:GOSUB 1050:UPDAT=C1
372 IF K=2 AND PAGE>C1 THEN PAGE=PAGE-
C1:GOSUB 1050
373 IF K=5 THEN GOSUB 1100:GOSUB 910:G
OTO 950
374 IF K=19 THEN 600
375 IF K=6 THEN GOSUB 910:GOTO 1300
376 IF K=18 THEN 700
377 IF K=12 THEN 1400
378 IF K=15 THEN 800
379 IF K=24 THEN 850
380 GOTO 310
399 REM * INPUT A NUMBER *
400 TRAP 550:POSITION COL,ROW:? NBLK$;
:POSITION COL,ROW:P=2:N$(1)=CHR$(K):?
CHR$(K);
402 POKE 752,C0:? " ";CHR$(126);
410 POKE 694,0:GET #1,K
420 IF K=155 THEN POKE 752,C1:? " ";G
OTO 500
430 IF K=126 AND P>C1 THEN P=P-C1:N$(P
)=? " ":CHR$(K);
440 IF P>15 THEN 410
450 IF (K)47 AND K<50 OR K=43 OR K=45
OR K=46 OR K=69 THEN N$(P,P)=CHR$(K):
? CHR$(K);:P=P+C1
460 GOTO 410
500 P=(PAGE-C1)*20+ROW:R$=N$:N=VAL(R$)
510 IF COL=XLT THEN X(P)=N:COL=VLT:GOS
UB 920:GOTO 530
520 IF COL=VLT THEN Y(P)=N:COL=XLT:GOS
UB 920:GOSUB 560
530 TRAP 40000:GOTO 310
550 POSITION COL,ROW:? NBLK$:TRAP 4000
0:GOTO 310
559 REM * ADVANCE ROW, CK FOR PAGE CRO
SSING *
560 GOSUB 910:IF ROW=20 AND PAGE=MPGS
THEN GOSUB 900:RETURN
570 ROW=ROW+1:GOSUB 910:IF ROW>20 THEN
ROW=1:PAGE=PAGE+1:GOSUB 1050:RETURN
580 GOSUB 900:RETURN
599 REM * SAVE A FILE *
600 GOSUB 1100
610 GOSUB 1200:OPEN #2,0,0,FNM$:PRINT
#2;ENTR
620 FOR J=1 TO ENTR
630 PRINT #2;MARK(J);",";X(J);",";Y(J)
640 NEXT J
650 CLOSE #2:GOSUB 1050:TRAP 40000:GOT
O 310
660 POSITION 0,21:? "**** DISK ERROR -
FILE NOT SAVED ****":GOSUB 1370:GOTO
650
699 REM * READ A FILE *
700 GOSUB 910:GOSUB 1100:GOSUB 1200:TR
AP 760:OPEN #2,4,0,FNM$:INPUT #2,NADD
705 IF ENTR+NADD>MPTS THEN POSITION 0,
21:? " **** FULL ****"
":GOSUB 1370:GOTO 750
710 FOR J=ENTR+1 TO ENTR+NADD
720 INPUT #2;M,X,Y
730 MARK(J)=M:X(J)=X:Y(J)=Y
740 NEXT J:ENTR=ENTR+NADD
750 CLOSE #2:GOSUB 1170:GOSUB 1050:TRA
P 40000:GOTO 310
760 POSITION 0,21:? " **** FILE
NOT FOUND ****" ":GOSUB 1370:GOT
O 750
799 REM * BUBBLE SORT OF X'S *
800 GOSUB 1100:? CHR$(125);" sorting
"
810 SFLG=0:FOR J=1 TO ENTR-1:IF X(J+1
)>X(J) THEN 830
820 TEMP=X(J):X(J)=X(J+1):X(J+1)=TEMP:
SFLG=1:TEMP=Y(J):Y(J)=Y(J+1):Y(J+1)=TE
MP
830 NEXT J:IF SFLG=1 THEN 810
840 GOSUB 1050:GOTO 310
849 REM * EXCHANGE X AND Y DATA *
850 GOSUB 910:? CHR$(125);" EXCHANGING
DATA"
852 POSITION 0,22:? "
"
860 GOSUB 910:FOR J=1 TO MPTS:TEMP=X(J
):X(J)=Y(J):Y(J)=TEMP:NEXT J:GOSUB 105
0:GOTO 310
899 REM * POSITION CURSOR ROUTINES *
900 PMC=PMOFF+4*(ROW+C1):FOR J=C0 TO 3
:POKE PMC+J,C255:NEXT J:RETURN
910 PMC=PMOFF+4*(ROW+C1):FOR J=C0 TO 3
:POKE PMC+J,C0:NEXT J:RETURN
920 POKE 53248,48+4*COL:RETURN
949 REM * UPDATE X'S BY EQUATION *
950 TRAP 990:? CHR$(125):IF ENTR=0 THE
N ? :? "WHAT MAXIMUM VALUE FOR ENTRIES
";:INPUT ENTR
951 IF ENTR>MPTS THEN 950
952 POSITION 2,4:? "980"? :? :? "CONT
"? :? " ENTER EQUATION (J,X,Y)"
954 POSITION 7,4
956 GET #1,K:IF K=155 THEN 960
958 ? CHR$(K);:GOTO 956
960 POSITION 0,2:POKE 842,13:STOP
970 POKE 842,12
975 FOR J=1 TO ENTR
980 X(J)=X(J)*3.14159
985 NEXT J
986 ? CHR$(125):TRAP 40000:GOSUB 1050:
GOTO 310
990 POSITION 0,21:? "**** ERROR-BE SURE
AND CHECK ENTRIES ****":GOSUB 1370:GOTO
986
999 REM * INITIALIZE ARRAYS ETC. *

```



```

1000 N9=3.3E-33:FOR J=1 TO MPT5:X(J)=N
9:Y(J)=N9:MARK(J)=0:NEXT J:ENTR=0:UPDA
T=0
1010 PAGE=1:ROW=1:XLT=7:COL=XLT:YLT=23
:JLT=1:MLT=5:C0=0:C1=1:C255=255:M1=-1:
M2=-2
1020 RETURN
1049 REM * DISPLAY A PAGE *
1050 ? CHR$(125);" PAGE ";PAGE;" 0
F ";MPG5:POSITION 25,C0:?"X":POSITION
33,C0:?"Y"
1051 STRT=(PAGE-C1)*20:FOR J=C1 TO 20:
POSITION JLT,J:?"STRT+J;" ":POSITION
MLT,J:?" "
1052 IF MARK(STRT+J)=M1 THEN POSITION
MLT,J:?"X"
1055 POSITION XLT,J:?" NBLK$:IF X(STRT+
J)<N9 THEN POSITION XLT,J:?" X(STRT+J)
1060 POSITION YLT,J:?" NBLK$:IF Y(STRT+
J)<N9 THEN POSITION YLT,J:?" Y(STRT+J)
1070 NEXT J
1079 REM * PRINT OPTIONS AT BOTTOM *
1080 POSITION 0,21:?" CTRL(arrows) M
k Del Undo Mxt Bk Xchg"
1082 POSITION 0,22:?"Rd Sav Odr Eqn E
SC DEL CLR Fil sTat Lst":GOSUB 900:GOS
UB 920
1090 RETURN
1099 REM * CLEAR UP BLANK SPACES *
1100 IF UPDAT=0 THEN GOSUB 1180:RETURN
1102 GOSUB 910:J=1:FF=0:?" CHR$(125);"C
LEANING UP DATA"
1104 POSITION 0,22:?"
";
1108 IF MARK(J)=M2 THEN J10
1110 IF (X(J)<N9 OR Y(J)<N9) AND J<M
PT5 THEN J=J+1:GOTO 1108
1120 FF=J:IF J=MPT5 THEN 1160
1140 IF (X(J)=N9 OR Y(J)=N9) AND J<MPT
5 THEN J=J+1:GOTO 1140
1142 IF MARK(J)=M2 AND J<MPT5 THEN J=J
+1:GOTO 1140
1144 IF J=100 AND (X(J)=N9 OR Y(J)=N9)
THEN 1160
1150 X(FF)=X(J):Y(FF)=Y(J):MARK(FF)=MA
RK(J):J=J+1:FF=FF+1:IF J<MPT5+1 THEN 1
140
1160 FOR J=FF TO MPT5:X(J)=N9:Y(J)=N9:
MARK(J)=0:NEXT J:ENTR=FF-1
1170 J=INT(ENTR/20):PAGE=J+1:ROW=ENTR+
1-(J*20):COL=XLT:GOSUB 900:GOSUB 920:U
PDAT=0:RETURN
1179 REM * COUNT NUMBER OF ENTRIES *
1180 J=1
1184 IF (X(J)<N9 OR Y(J)<N9) THEN J=J+
1:GOTO 1184
1186 ENTR=J-1:RETURN
1199 REM * INPUT FILENAME *
1200 GOSUB 910:POSITION 0,21:?"
INPUT FILENAME
";
1202 POSITION 0,22:?" D:
.DAT
";
1210 P=3:POSITION 12,22:FNMS="D:"
1220 GET M1,K
1230 IF K=155 THEN 1280
1240 IF K=126 AND P>3 THEN P=P-1:FNMS(
P)=" ":?" CHR$(K);
1242 IF K>96 AND K<123 THEN K=K-32
1250 IF ((K>47 AND K<58) OR (K>64 AND
K<91)) AND P<11 THEN FNMS(P,P)=CHR$(K)
:?" CHR$(K);P=P+1
1260 GOTO 1220
1280 FNMS(P)=DAT$:RETURN
1299 REM * SHOW .DAT FILES ON DISK *
1300 ? CHR$(125):OPEN #2,6,0,"D:*.DAT"
:TRAP 1340
1310 FOR K=0 TO 20 STEP 20
1330 FOR J=1 TO 22:INPUT #2;D$:POSITIO
N K,J:?" D$;NEXT J:NEXT K
1340 GOSUB 1370
1360 CLOSE #2:GOSUB 1050:GOTO 310
1370 POSITION 20,22:?"PRESS SPACEBAR"
;
1380 GET M1,K:IF K<32 THEN 1380
1390 RETURN
1399 REM * LIST VALUES ON PRINTER *
1400 ? CHR$(125):GOSUB 1100:NBR=INT(EN
TR/2):MARG=9-MDP:ODD=0:IF INT(ENTR/2)<
>ENTR/2 THEN NBR=NBR+1:ODD=1
1401 TRAP 1440:OPEN #3,8,0,"P:":CLOSE
#3
1402 LPRINT :LPRINT " #
X Y ";:LPRINT " #
X Y"
1404 LPRINT :FOR M=1 TO NBR:N$=STR$(M)
:L=LEN(N$):LPRINT NBLK$(1,4-L);M;" ";
1405 IF MARK(M)<M1 THEN LPRINT " ";:
GOTO 1410
1406 LPRINT "X ";
1410 N=X(M):IF N=N9 THEN LPRINT NBLK$;
:GOTO 1414
1412 GOSUB 1450:LPRINT NBLK$(1,MARG);N
$;
1414 N=Y(M):LPRINT " ";:IF N=N9 THEN L
PRINT NBLK$;" ";:GOTO 1418
1416 GOSUB 1450:LPRINT N$;NBLK$(1,MARG
);
1418 IF M=NBR AND ODD=1 THEN LPRINT :G
OTO 1434
1420 N$=STR$(M+NBR):L=LEN(N$):LPRINT M
BLK$(1,4-L);M+NBR;" ";
1422 IF MARK(M+NBR)<M1 THEN LPRINT "
";:GOTO 1426
1424 LPRINT "X ";
1426 N=X(M+NBR):IF N=N9 THEN LPRINT NB
LK$;" ";:GOTO 1430
1428 GOSUB 1450:LPRINT NBLK$(1,MARG);N
$;
1430 N=Y(M+NBR):LPRINT " ";:IF N=N9 TH
EN LPRINT NBLK$:GOTO 1434
1432 GOSUB 1450:LPRINT N$;NBLK$(1,MARG
)
1434 NEXT M
1436 TRAP 40000:GOSUB 1050:GOTO 310
1440 LPRINT :LPRINT :LPRINT "***** PRI
NTER ERROR *****":CLOSE #3:GOTO 1436
1449 REM * PUT # IN E FORMAT *
1450 NFLG=0:N$=STR$(M):IF N<0 THEN M$=
M$(2):NFLG=1
1460 J=1:DP=0:EX=0:FN0=1:EP=0:EPN=0:L=
LEN(N$)
1470 IF N=0 THEN 1550
1480 IF N$(J,J)="", THEN DP=J:J=J+1
1490 IF N$(J,J)="" THEN J=J+1:GOTO 14
80
1500 FN0=J
1510 J=J+1:IF J>L THEN 1550
1520 IF N$(J,J)="", THEN DP=J
1530 IF N$(J,J)="E" THEN EX=J
1540 GOTO 1510
1550 IF EX=0 THEN EP=VAL(N$(EX+1)):N$(
EX)="0000"
1560 IF LEN(N$)<20 THEN N$(LEN(N$)+1)=
"00000000000000000000000000000000"
1570 IF DP=0 THEN EPN=L-1
1580 IF DP>0 THEN EPN=DP-FN0:IF EPN>
0 THEN EPN=EPN-1
1590 N$(1)=N$(FN0):IF DP>FN0 THEN N$(
DP)=N$(DP+1)
1600 R$=N$(2):N$(2)=N$(1):N$(1,1)=""
:N$(3)="",N$(4)=R$:N$(NDP+4)="E":EP=EP
+EPN:N$(NDP+5)="+
1610 IF EP<0 THEN N$(NDP+5)="-"
1620 R$=STR$(ABS(EP)):IF LEN(R$)=1 THE
N N$(NDP+6)=""N$(NDP+7)=R$:GOTO 1640
1630 N$(NDP+6)=R$
1640 IF NFLG=1 THEN N$(1,1)="-"
1650 RETURN
1699 REM * STATISTICS *
1700 GOSUB 1100:?" CHR$(125):GOSUB 910:
TRAP 1800
1710 NBR=0:TOT=0:BIG=0:LITTLE=1.0E+97:
FOR J=1 TO ENTR
1712 IF Y(J)=N9 THEN 1730
1714 NBR=NBR+1
1720 TOT=TOT+Y(J)
1722 IF Y(J)>BIG THEN BIG=Y(J)

```


MAGIC 1 & 2

by Charlie Parker

XY ENTRY

```

1724 IF Y(J)<LITTLE THEN LITTLE=Y(J)
1730 NEXT J:MEAN=TOT/ENTR:TOT=0:MEANSQ
=MEAN*MEAN
1740 FOR J=1 TO ENTR:IF Y(J)<N THEN
TOT=TOT+Y(J)*Y(J)-MEANSQ
1742 NEXT J:VAR=TOT/(NBR-1):SD=SQR(VAR)
1750 ? :? :? "*****
***"? "STATISTICAL SUMMARY ***"?
*****
1760 ? :? :? "LARGEST VALUE = ";BIG:
?: "SMALLEST VALUE = ";LITTLE
1770 ? :? "MEAN VALUE = ";MEAN:?: "N
UMBER OF ENTRIES = ";NBR
1780 ? :? "VARIANCE = ";VAR:?: "STAN
DARD DEV. = ";SD
1790 TRAP 40000:GOSUB 1370:GOSUB 1050:
GOTO 310
1800 ? " ***** ERROR CHECK DATA VALUES
*****":GOTO 1790
2999 REM * PM GRAPHICS *
3000 GRAPHICS 7:POKE 106,PEEK(106)-5:P
M=PEEK(106)+1:POKE 54279,PM:GRAPHICS 0
:POKE 752,1:?" initializing"
3010 POKE 708,34:POKE 709,0:POKE 712,1
0
3050 POKE 704,56:POKE 559,46:POKE 5327
7,3:POKE 53256,3:PMOFF=(PM*2)*256+16
3069 REM * DISPLAY LIST *
3070 DLIST=PEEK(560)+256*PEEK(561):POK
E DLIST+3,198:POKE DLIST+6,134:FOR J=7
TO 26:POKE DLIST+J,130:NEXT J
3072 RESTORE 3074:FOR J=1569 TO 1586:R
EAD A:POKE J,A:NEXT J:POKE 512,34:POKE
513,6:POKE 54286,192
3074 DATA 180,72,173,33,6,73,2,141,33,
6,141,10,212,141,24,208,104,64
3079 REM * FLASHING CURSOR SEE COMPUTE
! DEC '81 *
3080 RESTORE 3082:FOR J=1536 TO 1568:R
EAD A:POKE J,A:NEXT J:A=USR(1536)
3082 DATA 104,169,17,141,40,2,169,6,14
1,41,2,169,30,141,26,2,96,173,243,2,41
,2,73,2,141,243,2,169,15,141,26,2,96
3090 RETURN

```

```

10 REM EXAMPLE #1 STRING MAGIC
50 DIM ROMCHR$(1):REM CREATE STRING FO
R ROM CHARACTER SET
70 DIM RAMCHR$(1):REM CREATE STRING FO
R RAM CHARACTER SET
90 DIM S1$(8),S2$(8),S3$(8),S4$(8):REM
STRINGS FOR NEW CHARACTERS (SQUARES)
100 REM LOAD BIT PATTERNS FOR SQUARES
110 FOR M=1 TO 8
120 READ C:S1$(M,M)=CHR$(C):NEXT M
130 DATA 255,129,129,129,129,129,129,2
55
140 FOR M=1 TO 8
150 READ C:S2$(M,M)=CHR$(C):NEXT M
160 DATA 0,126,66,66,66,66,126,0
170 FOR M=1 TO 8
180 READ C:S3$(M,M)=CHR$(C):NEXT M
190 DATA 0,0,60,36,36,60,0,0
200 FOR M=1 TO 8
210 DATA 0,0,0,24,24,0,0,0
220 READ C:S4$(M,M)=CHR$(C):NEXT M
230 REM SET UP PARMS FOR STRING MAGIC
240 ADDR=57344:REM ADDR OF ROM CHRSET
250 SIZE=1024:REM NEW SIZE OF VAR.
260 VNUM=0:REM USE 1ST VAR. (ROMCHR$)
270 REM CALL STRING MAGIC ROUTINE
280 GOSUB 580
290 REM MAKE ROOM FOR RAM CHRSET
300 RAMTOP=PEEK(106)-4
310 POKE 106,RAMTOP:GRAPHICS 2+16
320 REM SETUP PARMS FOR NEXT VAR.
330 ADDR=RAMTOP*256:REM RAM CHRSET
340 SIZE=1024:REM SAME AS ROM CHRSET
350 VNUM=1:REM SAME AS ROM CHRSET
360 REM CALL THE STRING MAGIC ROUTINE
370 GOSUB 580
380 REM MOVE ROM CHRSET TO RAM CHRSET
390 RAMCHR$=ROMCHR$
400 REM CHANGE ! CHAR TO A SQUARE
410 RAMCHR$(9,16)=S1$
420 REM POINT TO NEW CHRSET
430 POKE 756,PEEK(106)
440 REM PRINT SQUARES ON SCREEN
450 FOR M=1 TO 240
460 PRINT #6;"!";NEXT M
470 REM FLIP THRU DIFFERENT SQUARES
480 RAMCHR$(9,16)=S2$:GOSUB 560
490 RAMCHR$(9,16)=S3$:GOSUB 560
500 RAMCHR$(9,16)=S4$:GOSUB 560
510 RAMCHR$(9,16)=S1$:GOSUB 560
520 RAMCHR$(9,16)=S2$:GOSUB 560
530 RAMCHR$(9,16)=S3$:GOSUB 560
531 IF PEEK(764)<255 THEN POKE 764,25
5:RUN "D:MENU"
540 GOTO 480
550 REM SLOW DOWN SO WE CAN SEE IT

```

```

560 FOR M=1 TO 33:NEXT M:RETURN
570 REM STRING MAGIC ROUTINE
580 A=(ADDR-PEEK(140)+PEEK(141)*256):A
N=INT(A/256):AL=A-(AN*256):VUT=PEEK(13
4)+PEEK(135)*256:Q=VUT+VNUM*8+2
590 IF A<0 THEN A=PEEK(140)+PEEK(141)*
256:GRAPHICS 0:?"CAN'T GO BELOW ADDR
55":A:STOP
600 IF PEEK(Q-2)<128 THEN GRAPHICS 0:?"
VARIABLE #";VNUM;" NOT A STRING":STO
P
610 POKE Q,AL:POKE Q+1,AH:AH=INT(SIZE/
256):AL=SIZE-(AH*256):POKE Q+2,AL:POKE
Q+3,AH:POKE Q+4,AL:POKE Q+5,AH:RETURN

```

```

10 REM EXAMPLE #2 STRING MAGIC
50 DIM PM$(1):REM CREATE STRING FOR PL
AYER/MISSILES
70 DIM R1$(12),R2$(12):REM STRINGS FOR
DIFFERENT VERSIONS OF PLAYER (ROBOT)
80 REM LOAD BIT PATTERNS FOR ROBOTS
90 FOR M=1 TO 12
100 READ C:R1$(M,M)=CHR$(C):NEXT M
110 DATA 0,0,56,92,254,124,68,74,64,16
0,0,0
120 FOR M=1 TO 12
130 READ C:R2$(M,M)=CHR$(C):NEXT M
140 DATA 0,0,56,116,254,124,68,164,4,1
0,0,0
150 REM MAKE ROOM FOR P/M GRAPHICS
160 RAMTOP=PEEK(106)-4
170 POKE 106,RAMTOP:GRAPHICS 0
180 POKE 752,1:?"
190 SETCOLOR 2,0,0
200 REM SETUP PARMS FOR STRING MAGIC
210 ADDR=RAMTOP*256:REM SET TO PMBASE
220 SIZE=1024
230 VNUM=0:REM USE 1ST VAR. (PM$)
240 REM CALL STRING MAGIC ROUTINE
250 GOSUB 430
260 REM CLEAR P/M AREA
270 PM$(1,1)=CHR$(0):PM$(2)=PM$(1)
280 REM INITIALIZE P/M GRAPHICS
290 POKE 623,16:REM SET P/M PRIORITY
300 POKE 54279,RAMTOP:REM SET PMBASE
310 POKE 704,122:REM SET PLAYER COLOR
320 POKE 559,46:REM ALLOW DMA P/M GR.
330 POKE 53256,1:REM SET PLAYER SIZE
340 POKE 53248,120:REM SET PLAYER POS.
350 POKE 53277,3:REM TURN ON P/M GR.
360 P=636
370 SP=P-2:PM$(P,P+12)=R1$:GOSUB 410
380 SP=P-2:PM$(P,P+12)=R2$:GOSUB 410

```



LIBRARY LIST

PILOT DISK #3 (disk only)

RANDSPELL - For beginning spellers. You put the words on the screen and the child can study them as long as he wants. The words disappear and the child attempts to type them in correctly.

FAMILY SQUARE - A Tic-Tac-Toe type of game but you must answer questions before you are allowed to select your square. Game is not complete. You must supply the questions.

RHYTIME - Type in a word and then try typing in additional words until you find one that rhymes.

CHRUTIL - A PILOT character maker.

CHRRET - Utility character saver.

CHREX - An alternate character set.

LETTERMATCH - Match the lower case letter on the screen with the upper case from the keyboard.

Best of ACE, Vol. 9 (disk only)

NEWDOS - prints error messages on screen along with error number.

GLOBAL - Used with Filemanager 800 files.

FUNART - Used with the "Fun With Art" cartridge.

TOMB - Use your joystick to find the keys that open the treasure chamber in the Pharaoh's tomb.

KRAZY CLIMBER - Use your joystick to climb the outside of a building while dodging objects being dropped on you.

MX80DUMP - A revised graphics screen dump for the Epson MX80 printer.

PAINTBOX - An "Etch-a-sketch" type of program written in ACTION by Stan Ockers. No cartridge needed.

ALPHABEE - For the youngster learning the alphabet. Fly from flower to hive collecting the letters of the alphabet in the correct order. Written in ACTION by Stan Ockers. No cartridge needed.

STRINGMAKER - A machine language string maker written in BASIC.

FLASH - Numbers flash on the screen. A good way to speed up your reading and comprehension.

BULLANT - A 2 player game from Australia. Move your eggs from the bottom nest to the top.

SUPERLOADER - A method for loading assembler programs from within a BASIC program.

ACTION DISK #2 (disk only)

For the advanced programmer. ACTION CARTRIDGE REQUIRED.

SYMBOL TABLE - Two versions on this disk. Lists local variables by PROC/FUNC and globals at the end of compilation.

SCREEN DEMO - Just what it says.

CATCH - An error trapping routine that can be used in your ACTION programs.

TIME - A calender.

COLOR - Puts color bars on the screen.

TERMINAL - A modem program in ACTION. Documentation is on the disk.

PAINTBOX - An "Etch-a-sketch" type of program written in ACTION by Stan Ockers.

ACTION DISK #3 (disk only)

MAZE - Creates a maze that can be shown on the screen or dumped to a printer. Size is variable from 1 by 1 (not really practical) to 25 by 30. Also has an invisible option showing only the outline of the size.

ALPHABEE - For the youngster learning the alphabet. Fly from flower to hive collecting the letters of the alphabet in the correct order. Written in ACTION by Stan Ockers.

ESCAPE - Originally written in BASIC by David Bohlke.

Translated to ACTION by Bob Turner. A 3-D maze escape. Use the joystick to find your way out and the trigger to see a top view of the maze.

BOUNCE - A 2 player game from Stan Ockers. Try bouncing your man into the sky and landing on a cloud. Miss the trampoline and you end up in quicksand. If the bird flying around the screen catches you, you're a gonner.

PAGELIST - An ACTION utility to list an ACTION source code file to single sheets on an Epson printer.

TINTOWIZ - Converts TINYTEXT files to TEXTWIZARD format.

FILEINDEX (disk only)

A commercial quality disk inventory program 3 years in development. This makes keeping track of what is on which disk much easier. Complete documentation is on the disk.

SOURCE CODES DISK #2 (disk only)

Another collection of miscellaneous assembly and source codes from the newsletter. No documentation - not guaranteed to run. Some may be just "program modules" to be used within other programs.

ACE-C is here by JOHN PALEVICH
Modified by Ralph Walden
just \$10.00

ATARI COMPUTER ENTHUSIASTS EXCHANGE LIBRARY
374 BLACKFOOT STREET, EUGENE, OREGON 97404
(revised 12/1/84)

EXCHANGE POLICY

The ACE Exchange Library is a service designed to allow the sharing of user-written programs. There are two ways to acquire programs from the ACE exchange library: By ordering the disk(s) you desire and enclosing the appropriate amount of money, or by sending us programs you have written and wish to share with others, in exchange for your choice of any disk in the library. We strongly support this second way of obtaining programs, and will go out of our way to help you share your programs with other users. All disks (and tapes) in our exchange library are available for trade except SCOPY 810. We do ask that you play fair and do not send programs that are copyrighted unless you are the owner of the copyright and will agree to share your program(s) with others.

We will not exchange programs that duplicate existing programs in the library unless they are improved versions. A good example of this is TINYTEXT, which appears on several BEST OF ACE disks. The original program was written for us by Stan Ockers of Illinois, improved by Jim Carr of Oregon, further improved by Dale Lutz of Canada, improved by Bill Hardwick of Great Britain, and the latest version, an 80 column display by Neville Williams of Australia.

Programs that are submitted must be accompanied by an explanation of what the program does and a description of any "special" programming techniques used. We would also appreciate an accompanying article for the newsletter. Any Atari compatible word processor may be used or if you don't have one, try our own TINYTEXT (or just type it on your typewriter).

ACE is a nonprofit organization dedicated to helping its' members enjoy their Atari computers. Programs in the exchange library are contributed by members and other user groups. IT IS CONTRARY TO ACE BYLAWS TO EXCHANGE COPYRIGHTED PROGRAMS.

TO PLACE YOUR ORDER

All disks in the library are \$10 each single sided or \$15 double sided. We have had no problems with this in 4 years use.

Disks available for double or single density drives. Please specify double density with order. NOTE-do not order double (dual) density for the Atari 1050. If you

are using the 1050 drive, please order single density only.

All are disks listed are available on tape unless marked "disk only" next to the title.

To order any of the programs, send your check, payable to "ATARI COMPUTER ENTHUSIASTS", in the appropriate amount and your request to: Atari Computer Enthusiasts, 374 Blackfoot Street, Eugene Oregon 97404

In order to keep our costs down, we are not equipped to handle any credit card billings such as VISA or MASTER CHARGE.

Prices include postage in the USA and CANADA. FOR ORDERS TO ADDRESSES OUTSIDE THE USA AND CANADA, please add US \$2.00 per order (regardless of how many disks ordered) for airmail service.

OTHER ITEMS AVAILABLE DIRECTLY FROM OUR MEMBERS

Best of ACE books - bound editions of the ACE Newsletter.
Volume 1 covers all of the issues of the ACE newsletter from the first issue thru June of 1982.
Volume 2 covers July 1982 to June 1983.
Volume 3 covers July 1983 to June 1984. This is an excellent value. These books are only \$12.00 each (add \$2.00 extra per book for airmail postage). Order directly from:

Softfinder 1.2

Softfinder 1.2 is a composite index to Atari related articles in 5 popular periodicals from April 1981 thru June 1983. It is especially helpful for those of you who perhaps picked up a magazine off the news stand and typed in a program that does not run. The next issue of that magazine probably had the "fix" for that program. This will help you find it. Periodicals covered are ACE Newsletter, Analog Computing, Antic, Compute, and Creative Computing. The price \$7.00 postpaid in USA & CANADA, US \$8.00 elsewhere. Order directly from the publisher:

The above publications must be ordered directly from the publisher. DO NOT send your order to ACE.

Jim Carr, Valley Soft, 2660 S. W. DeArmond, Corvallis.

NEW DISKS FOR XMAS

Best of Ace #10 - Games

Best of Ace #11 - Utilities

STRING MAGIC

(reprint: Starfleet, November, 1984)

If you already know how to use Atari strings to create and modify character sets, move player-missiles smoothly and to perform animation by page flipping, then this article is not for you. However, if you want to know how to do these things (and more) then read on. The key to all this is a routine I call string magic. The routine allows you to modify the address and size of any string variable. So what? While this is no fantastic accomplishment, once you know how to do this a new whole world of "string magic" becomes available.

USING STRING MAGIC

To use the string magic routine you must first initialize the following variables:

ADDR — this parameter must contain the new address to which you want the string to point.

SIZE — this parameter must contain the new size of the string variable. Even if you are not changing the size, you must still initialize this variable to the desired size.

VNUM — this parameter must contain the variable number of the string to be modified. This tells the string magic routine which string variable is to have its address and size changed.

After these parameters are set to the desired values, then you can call the string magic routine simply by doing a GOSUB to the routine. The routine can be placed anywhere in your program since there are no GOTOs or GOSUBs in the routine. A note of caution — the variable numbers are assigned by BASIC as the variables are entered, not necessarily the order they appear in the program. For example, if you entered the Line 50 `B$=A$` before entering the Line 10 `DIM A$(40),B$(40)`, then the variable number for the string `B$` will be zero and the string `A$` will have a variable number of one (variable numbers start at zero). This is true even though the `DIM` statement lists the string `A$` first. If you are not sure of the order the variables were entered (and thus their variable numbers), then the easiest way to reorder (or renumber) them is to first `DIM` them in the order you want them with the `DIM` statement the first line in the program. Next `LIST` the program to disk or tape. Type in `NEW`, then `ENTER` the program from disk or tape. The sequence in which the variables is entered will now be the order in which you dimensioned them.

HOW IT WORKS

The string magic routine works by changing the Variable Value Table BASIC builds. By changing values in the table, we can change the address and size of any string. For you hackers there is a complete description of how the routine does this at the end of this article. For the rest of you who just want to use the routine, there are several things you should keep in mind. The routine can be used whenever you want to move large sections of memory around quickly (such as moving the standard character set from ROM to RAM) or when you want to change sections of memory quickly (as with player/missiles). For example, if you point the string `RAMCHR$` to the area of memory where you want the character set to be in RAM, then to move the character set from ROM to RAM can be accomplished simply by using the statement `RAMCHR$=ROMCHR$`. There are some limitations on the routine. The minimum address to which you can point a string is the address of the string/array area set up by BASIC. The string/array area is located immediately after your program. Thus, you cannot point a string to any memory address lower than the end of your program. If you try to set the address of a string to a lower memory address, then the routine will display an error message and abort. Also, you can use the routine only on string variables. If you try to change the address or size of a variable which is not a string, then the routine will display an error message and abort.

CHARACTER SET MAGIC

Example program 1 shows you how to use the string magic routine to move the standard character set from ROM to RAM, and how to use strings to quickly modify the new character set. Before entering the program, don't forget to type in `NEW`, and make sure you enter the lines in sequence so the variable numbers will be correct. When entering this and the other example programs, you may omit all the `REM` lines if you like since there are no `GOSUBs` to `REM` lines.

Take a look at example program number 1. The string `ROMCHR$` dimensioned on Line 50 will be modified to point the standard character set on ROM. You sharp eyed readers may have noticed something — the dimensioned size of the variable is only one. Why one instead of 1024 (the size of the full character set)? This is to save memory. Whenever you `DIM` a string, BASIC automatically reserves space in the string/array area for the maximum size of the string. However, since we are going to change the address of the string so it will no longer point to the string/array area, then why reserve room for the string? We must `DIM` the string for at least one or BASIC won't like it. The same is true of the `RAMCHR$` string. The string magic routine will be used to point this string to the area of memory where we want the new character set. The strings `S1$`, `S2$`, `S3$` and `S4$` will contain our new characters — different sizes of squares.

Lines 110-210 load in the different bit patterns forming the characters. If you don't understand how these bit patterns form the new characters then see *COMPUTE'S FIRST BOOK OF ATARI GRAPHICS*.

Lines 240-280 set up the string magic parameters for `ROMCHR$` and call the string magic routine. The `ADDR` parameter is set to 57344, the location of the standard character set in ROM. The `SIZE` parameter is set to 1024 — the size of the full character set, and the `VNUM` parameter is set to zero — the variable number for the string `ROMCHR$`.

Next, Lines 300 and 310 reserve four pages (1024 bytes) of memory at the top of RAM. This is where we will put our new character set. The `GRAPHICS` statement on Line 310 is necessary to force BASIC to move the display list and display memory under the new RAM top. Without this, our new character set will conflict with display memory.

The string magic parameters for `RAMCHR$` are set up next. The `ADDR` parameter points to the newly created space at the top of RAM and the `SIZE` parameter is set to the same size as for `ROMCHR$`. The `VNUM` parameter is set to one — the variable number for `RAMCHR$`. After calling the string magic routine we can now move the ROM character set to our reserved space in RAM simply by moving `ROMCHR$` to the `RAMCHR$`. Since the new character set is now in RAM and `RAMCHR$` points to it, we can now modify the character set by changing values in the string.

Line 410 makes the changes. It changes the exclamation point character (!) to our new character contained in the `S1$`. Don't forget, the character set is in internal character order. Table 9.6 in your BASIC Reference Manual lists the character order of the internal character set. To get the position of the bit patterns for a character in the character set, you must multiply the number of the desired character by 8, then add one. For example, to find the location of the bit pattern for the exclamation point in the string, multiply the character number of the exclamation point (one) by eight, then add one. This gives nine as the offset. Thus the bit pattern for the exclamation point is in position 9-16 of the string.

Line 430 points the system to our new character set and Lines 450-460 fill the screen with our new character — a square. Lines 480-530 rotate the exclamation point (now a square) through the different characters contained in `S1$`, `S2$`, `S3$` and `S4$` (our different sizes of squares). Note the `GOSUB 560` on each of these lines is to slow down the process so you can see it. To see how fast you can change characters using the string magic method, take out the delay by changing Line 560 to just a `RETURN`. This will allow the character to be modified at full speed.

PLAYER/MISSILE MAGIC

Program example #2 shows how you can use the string magic routine for player/missiles. Using a string for the player/missile graphics area allows the players to be moved quickly and smoothly simply by changing values in the string. On Line 50 `PM$` is `DIMed`. The string magic routine will be used to point this string to the player/missile graphics area. The strings `R1$` and `R2$` will contain the two versions of our player/missile robot. One version has the robot with the left leg up and the other version has the robot with the right leg up. By changing from one version to the other, we can make the robot appear to walk. The bit patterns for the two versions of our robot are loaded into `R1$` and `R2$` on Lines 90-140.

Note each version has two blank lines before and after the actual robot image. The zeroes at the beginning and end of the `DATA` statements create these blank lines. The blank lines are necessary to keep the player from leaving a "trail" as it is moved up or down. If you want to see this trail then change the `P=P-2` statement on Lines 370 and 380 to `P=P-3`. If you were going to actually move the player up or down three lines at a time then you would need three blank lines (three `DATA` statements of zero) at the front and back of the player and the size of `R1$` and `R2$` would have to be increased from 12 to 14.

The top of RAM is moved down 4 pages on Line 170 and a `GRAPHICS` statement is used to move the display list and display memory under this new top. We will use these 4 pages (1024 bytes) for our player/missile graphics area. Line 180 turns off the cursor and Line 190 sets the background color to black. The parameters needed for the string magic routine are initialized on Lines 210-230 and Line 250 calls the string magic routine. Now the string `PM$` points to the player/missile graphics area and the area can be modified simply by changing values in `PM$`.

Line 270 shows you a trick you can use to fill a string with a specific value. By placing a specific value in the first position of a string, then moving the value to position 2 through the end of the string, you can fill the entire string with the value. In this case position one of the string (the first position of the player/missile graphics area) is set to zero with the statement `PM$(1,1)=CHR$(0)`. Then by using the statement `PM$(2)=PM$(1)` we can clear the entire player/missile area of 1024 bytes to zero. This takes only a few micro seconds to perform as opposed to several seconds a `FOR/NEXT` loop takes.

After player/missiles are initialized on Lines 290-350 we are ready to place our robot on the screen and make him walk. This is done on Lines 360-400. By moving R1\$ and R2\$ through the PM\$ we are actually moving the player through the player/missile graphics area. The GOSUB 410 on Lines 370 and 380 is to slow the movement down so we can see it. to get the full effect of how quickly players can be moved using the string magic method, take out the delay loop on Line 410 by changing the line to 410 RETURN.

(Ed. note: There is another part to this about page flipping which will appear in the December issue of Starfleet)

— Charlie Parker

APPLIANCE INTERFACE

(reprinted from September, 1983 Computer Squad)

The Atari home computers are capable of doing much more than just playing games and using classic programming applications. The joystick ports each contain 16 lines which can be set to input or output, giving the user a maximum of 64,320 possible external control operations.

It is possible to use one of these lines set to output to turn off and on appliances, or to dial your phone under computer control.

The C.A.I. (computer appliance interface) consists of the following circuits:

- A. regulated 5vdc power supply
- B. two-stage inverting buffer
- C. opto-isolator (infra-red)
- D. single stage transistor amplifier
- E. micro relay
- F. macro relay

The power supply circuit (Figure 1) is constructed out of readily available components and is much larger than necessary to power the CAI. You may add 3 or 4 extra control line circuits and still have plenty of capacity left.

The main circuit (Fig. 2) shows the hookup of the two stage inverting buffer, the opto-isolator, and the amplifier powering the micro relay. All components listed are from Radio Shack, and readily available. the opto-isolator (infra-red) acts as a physical separation between the computer and the 110 oval being controlled. This provides the best protection possible so if anything were to go wrong, the computer won't get zapped with 100 vac.

The micro relay specified is manufactured by Allied, Inc., and available at 1355 N. McLean Blvd., Elgin, IL. You can substitute any 5vdc relay which pulls 50 ma or less. The Allied relay can be inserted into an IC socket, making construction easier.

A 9-pin connector connects the CAI to the computer. There are several you can use (Atari De9, with De110963-1 shell, or Radio Shack 276-1538, with 276-1539 shell). Soldering these pins to your connector cable and installing in the shell requires a great deal of patience and prior experience. Fig. 3 shows the pin outs to use. The device we built uses bit Line 3.

By directing the CAI circuit and adding to the control program, you can use bit lines 0,1,2 providing independent control of four separate devices. The power supply is more than adequate to power four control line circuits.

Finally, before you attempt to construct the circuit, beware of the following warnings (if these don't scare you off, heat up your soldering iron and go to it!):

1) Hooking up any non-standard device to your Atari will void your warranty.

2) Never attack a joystick controller to the connector ports when you are running the CAI program. Moving the stick to the "left" position could short a line and burn out the PIA chip.

3) Check your circuit for shorts, bridges, incorrect hook-ups, etc. before powering the device.

Ok, the warnings didn't scare you off. The circuit is built and you are ready to type in the driver program and try it. But... DON'T!! Before you expose your expensive and delicate computer to this strange device, let's bench test the operation to be sure the power drain on bit line 3 or the computer power supply does not exceed specs.

To bench test your device, you will need a Simpson 260 test meter or its equivalent. first set the test meter on 100 milliamp scale and connect the meter in series from the + 5vdc on your power supply to pin 7 on the connector. Now connect a shorting line from pin 4 and be able to touch the other end to the ground on the power supply. When inserting wires into the connector, be careful not to damage the receptacles by using excessive force or too large a wire.

Touch the shorting line to ground. You should hear the macro relay pull in, and your meter should read less than 50 milliamps. Next remove the test meter. Connect the + 5vdc directly to pin 7, reconnect the test meter (now in the 10 milliamp scale) in series with your shorting line and connect to ground. You should, once again, hear the macro relay pull in and your meter should read less than 1.6 milliamps. If either of these tests don't produce the amperage values expected, or the relay does not pull in, something is wrong. STOP and check for shorts, current wiring to transistors, etc. Do not try it on the computer until the above tests check out ok!!

When the device bench checks out satisfactorily, type in the driver program (Fig. 4), connect the CAI, set for appliance and try it.

Now the hard part is done, let's proceed to the auto phone dialer. Connect the phone to the CAI (fig. 5). Next type in the phone dialer program (Fig. 6), remove the phone from its cradle, and input a phone number.

This article only scratches the surface of what can be done to interface your Atari computer to the outside world. Other possibilities include adding a 4-16 line decoder to allow control of up to 16 devices using one port plug for remote control of appliances in your home.

— Jeff McHie

SYNSTOCK

Synstock (by Steven S. Weston, Synapse Software \$40) is another fine endeavor by the Synapse people. This program allows you to utilize your home computer to prepare stock price charts and to analyze stock trends. It is fully compatible with many of the other fine "SynApps" series, notably SynCalc and SynGraph, and allows you to easily convert current files in Atari's Stock Charting to SynStock.

As usual Synapse has provided a comprehensive 74 page instructional book, complete with tutorial. The chart produced by Synstock allows you to see daily highs, lows and closing volumes. It also allows you to do technical analyses in the form of moving averages and oscillators.

Synstock has the capability to create complete portfolios on your data discs. Each disc may hold up to 50 stocks with up to 300 days of quote data for each stock. However quote data is limited to a volume less than 9,999,999. With high, low, and close prices less than 999,999 and greater than 0.

Another nice feature of SynStock is its ability to down load quotes from CompuServe. For this you need a standard telephone Modem, at least 300 or 1200 baud and a Compuserve I.D. password. You are also required to use an ATARI 850 Interface Module and an 80 column printer using the Parallel Interface on the 850, with bit-image graphics capability. This service allows you to quickly create and update your files.

SynStock also allows you to produce hard copies of any of your charts. But it should be noted the types of printers which will fully support this printing are severely limited. You might wish to check and see if your printer is compatible first, if hard copies are a necessity. The only other warning which must go with this program is you must have an understanding of the stock market first. SynStock does not teach you about the stock market but does give you a valuable tool to expand the knowledge you have. This has nothing to do with how easy it is to understand the program itself. Synapse has done a fine job of making this a very user friendly program.

— Nick Chrones

TEE OFF

This is a quite addictive golf game which looks good. The player may choose from among five different courses. The structure of the code permits easy expansion of the number of courses. Blue = water; Yellow = sand trap; White = ball; Black = hole; Light Green = rough & trees; Dark Green = greens; Pink = flag.

The ball, hole and flag appear after selecting the course and pressing START. With the player as reference point, D is the direction towards which the wind is blowing. H is the number of hits so far. Angle is the angle at which you wish to hit the ball. The compass has 0 degrees at the top, 180 degrees at the bottom, etc. Force is the force with which you hit the ball, from 0 to 135.

The ball has a realistically calculated bounce which can bound over water or sand. The ball will stop dead if it hits sand, and will slow down in the rough. You must hit harder to get out of the sand. If you lose the ball, you are sent back to the tee.

— Sidney Brown

SPACE WASTE RACE

"An Animated Storybook" for Atari 800 and XL Series, published by Sunburst Communications, Inc. Designed for use by grades Pre-K to 2.

A REVIEW BY MILES ERICKSON, age 6½

When you boot up SPACE RACE, you get a menu that says:

```
CHOICES
STORYBOOK
LOOK, NOD & SHAKE
FUNTIME
END
```

THE STORYBOOK shows pictures and has some words about another moon fighting with earth's moon. The other moon wants garbage in space. Earth's moon doesn't want garbage in space.

LOOK, NOD, AND SHAKE would be fun for little kids. You can make a moon look different directions, and nod or shake its head.

FUNTIME has lots of choices:

MOONDROPS, FALL OUT, MOON PASS, HOLE IN THE MOON, MOON LOOKS, AND SCORE. MOON DROPS is the hardest. You have to count drops falling off of the moon and then type the right number. Other games are easier. Sometimes you have to know numbers, letters, directions like above, below, up, down, and right and left.

SPACEWASTE was easy to use. I didn't look at the manual at all. Little kids would have to have help. It was so easy. I think it is best for kids age 4-6, not 4-8. The pictures are very good. I liked it, and I think most people would like SPACEWASTE.

Dr. WACKO

Dr. C. Wacko's Miracle Guide To Designing Your Own Atari Computer Arcade Games is a book/disk package for any Atari computer. The Guide is aimed at people who have visions of creating a million dollar arcade game. Sound like you? Maybe I should put it differently. This package is ideally suited for people who have an adequate understanding of Atari BASIC, and want to learn the fundamentals of arcade game design and programming in a humorous way. There, now, doesn't that sound more like you?

The book is a spiral bound paperback with a colorful cover which will draw attention from 50 feet away. Enclosed in a packet on the inside of the front cover is the Wacko Disk. The Wacko Disk is a collection of all the listed programs in the book, saving you the trouble of typing them in. Throughout the book are many comical illustrations emphasizing the text and relieving some of the seriousness you might find in other such books. Several hilarious characters are introduced at the start of the book, and are to reappear later in the drawings.

The steps to designing an arcade game are thoroughly discussed. Topic areas include: Game design elements, graphics modes, customizing the character set, character animation and movement, joystick control, sounds, and player-missile graphics. Powerful things can be done with these techniques, as is demonstrated in Myrtle the Turtle, an example game which uses most of the techniques.

The Wacko Disk is very well set up. The menu corresponds to sections of the book, and the programs in those sections can be called up from the joystick controlled AUTORUN menu, making life much, much easier on the slow typist.

I found this package to be an excellent buy for the beginning gamer who wants to experiment with arcade-type programming.

Yet even this book doesn't lack faults. I felt some things weren't explained enough. The method is given very nicely, but often the WHY of things isn't explained quite clearly. I believe the author assumed the reader will know the inner workings of the Atari well enough to know such things.

Don't be misled by the title, either. This is not a good investment for an advanced BASIC graduate who is looking for a way to create a marketable machine language game. The Myrtle the Turtle game could not be compared to a popular machine language game. It's a good programming feat for a beginner to see. As I've said earlier, however, this is a GREAT investment for beginners of all ages and computer interests.

This is an excellent \$24.95 value, and I readily recommend it to anyone who falls in the category of beginning arcade game programmer.

— Tim Ebling

Walden Teaches Assembly Language

(In the last issue, Ralph began his column with the article on the 65C02, but his byline was left off. He is working hard on the "C" for ACE, and should be ready soon —ed.)

The normal method for accessing one of the handlers (keyboard, printer, screen, etc.) from assembly language is to call the CIO utility at \$E456. However, this requires a previously opened IOCB, and involves a fair amount of overhead in terms of memory usage. For example, one could get a key from the user with the instruction:

```
JSR $F6E2 ;(400 & 800's only!) The keyboard handler ignores all the
information the CIO hands it and just returns the next key pressed via
the A register. The problem with this is it is an "illegal" call, which
means it won't work on an XL computer. However, there is a method
for doing this legally to work on all the Atari computers. The way it
works is to get the address of the routine we want from its legal loca-
tion, push the address onto the stack, and do a return, which will ef-
fectively jump us to the address. Note the RTS at the end is nec-
essary, and the routine jumped to will return via its own RTS.
```

```
0100 GETKEY LDA $E425
0110 PHA
0120 LDA $E424
0130 PHA
0140 RTS
```

A similar method can be used to send something to the screen: (the A register contains the character to send)

```
0100 PUTEDIT TAY
0110 LDA $E407
0120 PHA
0130 LDA $E406
0140 PHA
0150 TYA
0160 RTS
```

One can also use this method to output a byte to a file which has been opened. When CIO opens an IOCB, it puts the PUTBYTE address in \$346,X (where the X register equals the IOCB times 16). We can use that address as a way to bypass CIO for a much faster way of outputting a byte of information. There is one exception, though. The XL operating system broke Atari's own rules when writing the printer handler, so you will have to store the byte into \$342,X (the command location) before calling the printer handler. Here is the routine for outputting to a device/file:

```
0100 PUTBYTE TAY ;hold the byte while we setup
0110 LDA $0347,X ;X = IOCB*16
0120 PHA
0130 LDA $0346,X
0140 PHA
0150 TYA
0160 RTS
```

— by Ralph Walden



Atari Computer Enthusiasts

A.C.E. is an independent, non-profit and tax exempt computer club and user's group with no connection to the Atari Company, a division of Warner Communication Company. We are a group interested in educating our members in the use of the Atari Computer and in giving the latest News, Reviews and Rumors.

All our articles, reviews and programs come from you, our members.

Our membership is world-wide; membership fees include the A.C.E. Newsletter. Dues are \$12 a year for U.S., and \$22 a year Overseas Airmail and include about 10 issues a year of the ACE Newsletter.
Subscription Dep't: 3662 Vine Maple Dr., Eugene, OR 97405.

President— Robert Browning, 90 W. Myoak Dr., Eugene, OR 97404
503-689-1513
Vice Pres— Larry Gold, 1927 McLean Blvd., Eugene, OR 97405
503-686-1490
Secretary— Bruce Ebling, 1501 River Loop #1, Eugene, OR 97404
503-688-6872
Librarian— Ron and Aaron Ness, 374 Blackfoot, Eugene, OR 97404
(503)689-7106
Editors—
Mike Dunn, 3662 Vine Maple Dr., Eugene, OR 97405 503-344-6193
Jim Bumpas, 4405 Dillard Rd., Eugene, OR 97405 503-484-9925
E.R.A.C.E. (Education SIG Editor)—
Nick Chrones, 2333 Monroe St., Eugene, OR 97405 503-343-6966

Send 50c stamps or coin (\$1 overseas) to the Ness' for the new, updated ACE Library List-new in May 84 !

Best of ACE books

Volume 1 contains bound issues of the ACE Newsletter from the first issue, Oct 81 to June of 1982

Volume 2 covers July 1982 to June 1983

Only \$12 each (\$2 extra for Airmail). Available only from:

George Suetsugu
45-602 Apuapu St
Kaneohe, HI 96744

SortFinder 1.2

A composite index of Atari related articles from 5 popular computer periodicals from Apr '81 to June '83, including ACE. Only \$6 for ACE member from:

Jim Carr, Valley Soft
2660 S.W. DeArmond
Corvallis, OR 97333

TYPESETTING FROM YOUR COMPUTER

ATARI OWNERS: If you have a modem, text editor, and communications program to send ASCII files, you should consider the improved readability and cost savings provided by **TYPESETTING** your program documentation, manuscript, newsletter, or other lengthy text instead of just reproducing it from line printer or daisy-wheel output. Computer typesetting by telephone offers you high quality, space-saving copy that creates the professional image you want! Hundreds of type styles to choose from with 8 styles and 12 sizes "on line." And it's easy to encode your copy with the few typesetting commands you need.

COMPLETE CONFIDENTIALITY GUARANTEED

— Bonded for your protection —

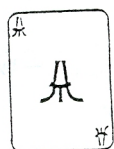
PUBLICATION DESIGN, EDITING, & PRODUCTION

Editing & Design Services Inc.

30 East 13th Avenue Eugene, Oregon 97401
Phone 503/683-2657

Bulletin Board . . . (503) 343-4352

On line 24 hours a day, except for servicing and updating. Consists of a Tara equipped 48K Atari 400 with a TARA keyboard, 2 double-density double sided disk drives, an Epson MX80 printer, a MicroBits 1100E Modem; running the FoReM Bulletin Board software distributed by MicroBits.

 **ATARI
COMPUTER
ENTHUSIASTS**
3662 Vine Maple Dr. Eugene OR 97405

FIRST CLASS MAIL